METHOD AND SYSTEM TO RECOVER A FAILED FLASH OF A BLADE SERVICE PROCESSOR IN A SERVER CHASSIS

BACKGROUND OF THE INVENTION

1. Technical Field

[0001] The present invention relates in general to the field of computers, and in particular to multiple blade servers housed in a server chassis. Still more particularly, the present invention relates to a method and system for automatically recovering a failed flash of a blade service processor.

2. Description of the Related Art

[0002] Server blade computers offer high-density server boards (blades) in a single chassis (blade chassis). A typical server blade computer is illustrated in **Figure 1**, identified as server blade chassis **102**. Server blade chassis **102** includes multiple hot-swappable server blades **104a-n**. There are typically fourteen server blades **104** in server blade chassis **102**. The operations of server blades **104** are coordinated by logic identified as a management module **108**, which typically includes a processor for controlling input/output (I/O) functions, interfacing with a network **106** (such as the Internet or a Local Area Network), and allocating jobs and data to the different server blades **104**.

[0003] Another function of management module **108** is to program Flash Read Only Memory (Flash Memory) in server blades **104**. This flash operation updates firmware in the server blade **104**, resulting in optimized operation. However, since server blades **104** are hot-swappable, there is usually nothing to prevent an engineer from unwittingly removing a server blade **104** from a mid-plane or back-plane (not shown) of server blade chassis **102** while the server blade **104** is in the middle of a flashing operation, which can take several minutes. When the partially flashed server blade **104** is re-installed into server blade chassis **102** or another chassis, it will often malfunction. Upon being re-installed into server blade chassis **102**, self-diagnostic logic in the re-installed server blade **104** will recognize that the flash operation failed

to fully execute. However, the server blade **104** will often be crippled to the point of not knowing its bus address or physical location within server blade chassis **102**, and thus unable to advise management module **108** of the problem with the aborted flash.

[0004] Similarly, even if server blade **104** is not removed from server blade chassis **102**, but the flashing operation fails, management module **108** will likely not know of the failure. Again, server blade **104** will be unable to notify management module **108** of the problem.

[0005] What is needed, therefore, is a method and system for enabling a server blade to communicate with a management module in a server blade chassis after a failed flash operation, which resulted in the server blade not knowing its location in the server blade chassis.

SUMMARY OF THE INVENTION

[0006] The present invention is directed to a method and system for recovering a server blade, in a multiple server blade computer, that is lost during a flash update operation on a service processor in the server blade. Because of the flash update failure, the lost server blade is unaware of its location on a management pathway, which in an exemplary form may be a midplane or a bus, which connects the server blade and a management module in a server blade chassis. The lost server blade puts a signal on the management pathway indicating that the flash failed. The signal is put on a special channel reserved for such messages. The management module receives the signal, and then determines which of the multiple server blades are lost due to the flash update failure.

[0007] In the event of multiple flash update failures occurring contemporaneously, the management module blocks communication with all lost server blades except one, and re-flashes the one unblocked server blade, thus enabling that one unblocked server blade to define its location (address) on the management pathway. Each of the other lost server blades are sequentially unblocked, re-flashed, and thus enabled to define their locations on the management pathway.

[0008] The above, as well as additional objectives, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further purposes and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, where:

[0010] Figure 1 depicts a prior art diagram of a server blade chassis coupled to a network;

[0011] Figure 2a illustrates a blade chassis incorporating a blade blocker associated with a management module for isolating and locating a server blade that is lost due to a flash failure;

[0012] Figure 2b depicts an organization of a non-volatile memory in a service processor in one of the service blades;

[0013] Figures 3 is a flow-chart of steps taken to retrieve a single lost server blade; and

[0014] Figure 4 is a flow-chart of steps taken to retrieve more than one lost serve blade.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

[0015] With reference now to **Figure 2a**, there is depicted a schematic block diagram of a server blade chassis **200** according to a preferred embodiment of the present invention. For the sake of clarity, only one management module **202** and three server blades **204a-n** are depicted. However, in a preferred embodiment, a backup management module **202** (not shown) is incorporated into server blade chassis **200**, and server blade chassis **200** has a midplane **206** capable of connecting fourteen or more server blades **204**.

[0016] Management module **202** is a logic capable of managing multiple server blades **204**. Management module **202** is coupled to server blades **204a-n** via a management pathway identified as midplane **206**. Midplane **206** is a backplane, mounted in the middle of server blade chassis **200**, that contains circuitry and sockets into which additional electronic devices or cards, including server blades **204** can be plugged. Midplane **206** contains at least one bus for secure internal communication between management module **202** and server blades **204a-n**, as well as between and among server blades **204a-n** themselves, via respective service processors **208a-n**.

[0017] Management module **202** is capable of detecting the presence, quantity, type and revision level of each server blade **204**, power module **210**, and midplane **206** in the system. Management module **202** can also direct flashing operations and identify failures of flashing operations, as described further below.

[0018] Each server blade **204** has at least one central processing unit (CPU) **212**, and a non-volatile memory (NVM) **214**. Preferably, NVM **214** is a Flash Read Only Memory ("Flash ROM" or "Flash Memory"), which can be erased and reprogrammed in units of memory called blocks. NVM **214** may also include non-volatile Electrically Erasable Programmable Read Only Memory (EEPROM), which is similar to Flash Memory except that EEPROM is erased and rewritten at the byte level, and is usually smaller in capacity than the flash memory.

[0019] When a server blade **204** is shipped from a manufacturer, the NVM **214** is typically pre-burned with firmware, including a Basic Input/Output System (BIOS) as well as software for monitoring the server blade **204**. Such monitoring may include regulating operating temperatures via speed adjustments to cooling fans **215**, controlling Direct Access Storage Devices (DASD's), monitoring and controlling voltages throughout the system, determining the power-on status of the server blade **204**, requesting access to a shared keyboard, video, mouse, Compact Disk-Read Only Memory (CD-ROM) and/or floppy disk drives, as well as monitoring the Operating System (OS) running on the server blade **204**. In order to take advantage of updates and other optimizations, this firmware is periodically updated by management module **202**, which re-flashes the firmware updates into NVM **214.**

[0020] For example, an updated flash code **216** accessible to (although not necessarily within) management module **202** can be downloaded to any or all service processors **208**. Each service processor **208** controls the flashing of the flash code **216** into its respective associated NVM **214**. If the flashing of flash code **216** into NVM **214** fails, then management of server blade **204** may be lost.

[0021] For example, consider the following example of a failed flashing operation of updated flash code **216** into NVM **214a**. During the flashing operation, server blade **204a** may be physically removed from its slot before the flashing operation completes. Likewise, there may be an intermittent power failure or spike during the flashing, or there may simply be a control, timing or any other software or hardware error that causes the flashing to fail to complete. Such a failure may result in server blade **204a**'s service processor **208a** "forgetting" its address on midplane **206**, and thus the address of server blade **204a**, as described below.

[0022] With reference now to **Figure 2b**, NVM **214** is divided into two sections: a protected area **218** and a flashable area **220**. Stored within protected area **218** is non-erasable (not capable of being overwritten) code, which may comprise Vital Product Data (VPD) such as the serial number, model number and Universal Unique IDentifier (UUID) of the server blade **204** associated with that NVM **214**. Protected area **218** may also include tracking data, including

which other server blade chassis **200** the server blade **204** has been mounted to in the past, if any. However, protected area **218** does not include the current address ("management pathway identity") of the server blade **204** on management pathway illustrated as midplane **206**.

[0023] Midplane **206** contains sockets **222** into which server blades **204** can be plugged. When a server blade **204** is plugged into a specific socket **222**, a physical address is established for that server blade **204**. For example, consider server blade **204a** being plugged into socket **222a**. A control logic, depicted as I^2C logic **224a**, which is compliant with the Phillips' Inter-IC (Integrated Circuit) standard (incorporated by reference in its entirety herein and commonly referred to as "I^2C"), detects the presence of server blade **204a** in socket **222a**. I^2C logic **224a**, operating in conjunction with management module **202**, assigns a physical address on a bus in midplane **206** to server blade **204a** when server blade **204a** is plugged into socket **222a**. Preferably, each server blade **204** is associated with a unique I^2C logic **224**, which is preferably connected to midplane **206** as depicted in **Figure 2a**. Alternatively, a single I^2C logic **224** can be used by all server blades **204**.

[0024] Alternatively, each socket blade **204** may have a unique Internet Protocol (IP) address on midplane **206**. That is, midplane **206** may support intercommunication using IP addressing protocol, in which each device connected or coupled to midplane **206** contains an IP address assigned by logic (not shown) that is either within or outside server blade chassis **200**. For example, a Dynamic Host Configuration Protocol (DHCP) server **110**, as shown in **Figure 1**, may be used to assign an IP address to server blade **204a**. Communication with server blade **204a** is thereafter via a Network Interface Card (NIC) **226a** that is associated with server blade **204a**.

[0025] Management module **202** manages a blade address list **228**, which is a list of all management pathway locations (either a physical bus address if socket **222** is used or an IP address if NIC **226** is used) on midplane **206**. This blade address list **228** is used to identify a lost server blade **204** in steps described in **Figures 3** and **4**.

[0026] With reference again to **Figure 2a**, situated, in an exemplary manner, between management module **202** and midplane **206** is a blade blocker **230**, which selectively blocks communication between management module **202** and any combination of specified service processors **208** in server blades **204**. A default state of blade blocker **230** allows unrestricted communication between management module **202** and server blades **204** (via respective service processors **208**), with specific communication between management module **202** and server blades **204** being blocked upon a signal/command from management module **202**. Blade blocker **230** is used to isolate a specific server blade's **204** service processor **208** when multiple server blades' **204** service processors **208** are lost, as described in **Figure 4**.

[0027] Also associated with midplane **206** is a presence detect device **232**. Presence detect device **232**, which may be an I^2C device, is able to communicate with management module **202** to identify which server blades **204** are installed on the midplane **206**.

[0028] With reference now to **Figure 3**, there is depicted a flow-chart of steps taken in a preferred embodiment of the present invention to recover a lost server blade. Starting at initiator **302**, Flash ROM in a service processor of a server blade is flashed with an update of firmware (block **304**). This firmware, downloaded from the management module to the service processor, is preferably that described above in reference to **Figures 2a-b**. That is, this firmware, which is downloaded (flashed) into the flashable area of the Flash ROM, is an update of the previously stored firmware, and includes code that optimizes the operation of the associated server blade.

[0029] During the flash operation, the logical location of the server blade may become undefined until the flash operation is successfully completed. The flash operation may not successfully complete, due to reasons discussed above. If not (query block **306**), the server blade transmits an error signal (block **308**). This error signal is transmitted on a special address (physical wire or IP address) in the midplane between the management module and the server blade. For example, if the server chassis has slots for fourteen server blades, each having a unique address on the midplane bus, then a fifteenth address on the midplane bus is reserved for error signals indicating a failed flash operation of the Flash ROM in a server blade.

[0030] However, the management module will not know just from the error signal alone which server blade failed to flash its Flash ROM with the updated firmware. Therefore, the management module first reviews a list of locations (addresses) of all server blades identified as being part of the server chassis system (block **310**). The management module then determines which server blade is "lost" (due to having its location undefined during the failed flash operation) by comparing locations on the server blade address list (of all server blades that should be on the system) with the addresses of server blades that have currently communication ability with the management module. This communication ability can be achieved by the management module by monitoring traffic to and from the server blades and using the presence detect device **232** as described above in reference to **Figure 2a**. The server blade that is on the list but not communicating with the management module is deduced to be the lost server (block **312**).

[0031] The management module then provides (block **314**), preferably using the I^2C logic **224** described in **Figure 2**, the server blade its current management pathway location (physical location or IP address). This current location identity is preferably provided by translating the current location identity from the blade address list **228** shown in **Figure 2a**. Now that the management module and server blade knows the restored location identity of the server blade, the management module can re-flash the Flash ROM in the server blade (block **316**), and the process ends (terminator block **318**).

[0032] There may be instances in which two or more server blades fail. The steps shown in **Figure 4** address a preferred method for handling this condition. While the steps are described as only two server blades failing to properly flash new firmware, the method is appropriate for any number of server blades, as described herein.

[0033] Starting with initiator block **402**, the management module flashes the Flash ROMs of a first and second server blade (block **404**). If they both fail (query block **406**), then both server blades will be lost. One (or preferably both) of the server blades put a failure signal on the midplane bus connected to the management module, which receives the failure signal (block

408). If both the first and second server blades send a failure signal, then the management module may receive duplicate failure signals, indicating two failed flashes. Alternatively, the management module may receive only a single failure signal, for both server blades, whether both server blades or only one server blade sent the failure signal.

[0034] The management module cannot communicate to both lost server blades, due to collision problems that would occur if both server blades were at the special address at the same time. Therefore, the management module must first block out one of the server blades (block **410**) using the blade blocker described in **Figure 2a**. In the same manner described in **Figure 3**, the management module then re-establishes a location identity for the unblocked server blade and re-flashes that server blades Flash ROM (block **412**). Next, the management module directs the blade blocker to unblock the blocked server blade, the identity location for the other lost server blade is established, and its Flash ROM is re-flashed (block **414**) in a same manner described above, thus ending the process (terminator block **416**).

[0035] If there are more than one server blades that have been lost due to a failed flashing of firmware to the Flash ROM, then the blade blocker continues to block all but one server blade at a time as each server blade's identity location is re-established.

[0036] The present invention thus provides a reliable method and system for recovering lost server blades that have been lost due to failed flashes. By identifying the lost server blades, the updated firmware for the server blade can be re-flashed, allowing the server blade to operate at maximum efficiency.

[0037] It should be understood that at least some aspects of the present invention may alternatively be implemented in a program product. Programs defining functions on the present invention can be delivered to a data storage system or a computer system via a variety of signal-bearing media, which include, without limitation, non-writable storage media (e.g., CD-ROM), writable storage media (e.g., a floppy diskette, hard disk drive, read/write CD ROM, optical media), and communication media, such as computer and telephone networks including Ethernet.

It should be understood, therefore in such signal-bearing media when carrying or encoding computer readable instructions that direct method functions in the present invention, represent alternative embodiments of the present invention. Further, it is understood that the present invention may be implemented by a system having means in the form of hardware, software, or a combination of software and hardware as described herein or their equivalent.

[0038] While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.